

[illegible]


```
1 0001 0 MODULE RPG$IOEXCEPTION( %TITLE, 'Signal I/O errors'
2 0002 0 IDENT = '1-003' ! file:RPGIOEXCE.B32 EDIT:DG1003
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1 ++
30 0030 1 FACILITY: RPGII SUPPORT
31 0031 1
32 0032 1 ABSTRACT:
33 0033 1
34 0034 1 This procedure is called to process a wide variety of I/O
35 0035 1 exceptions. It retrieves the file name string and signals the errors.
36 0036 1
37 0037 1
38 0038 1 ENVIRONMENT: Vax-11 User Mode
39 0039 1
40 0040 1 AUTHOR: Debess Grabazs, CREATION DATE: 18-Jan-1983
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. DG 18-Jan-1983
45 0045 1 1-002 - Reference to RPG$_INVDATTYP taken out.
46 0046 1 Reference to RMSS$_KRF changed to RMSS$_REF. DG 11-Jul-1983
47 0047 1 1-003 - Remove mapping of RMSS$_KEY to RPG specific error. DG 10-Aug-1983
48 0048 1 --
49 0049 1
50 0050 1 !<BLF/PAGE>
```

```
.. 52      0051 1 %SBTTL 'Declarations'
.. 53      0052 1 | +
.. 54      0053 1 | | PROLOGUE FILE:
.. 55      0054 1 | | -
.. 56      0055 1 | |
.. 57      0056 1 | REQUIRE 'RTLIN:RPGPROLOG';
.. 58      0121 1 | | Switches, PSECTs, macros,
.. 59      0122 1 | | | linkages and LIBRARYs
.. 60      0123 1 | | +
.. 61      0124 1 | | | LINKAGES
.. 62      0125 1 | | | NONE
.. 63      0126 1 | | | -
.. 64      0127 1 | | |
.. 65      0128 1 | | | +
.. 66      0129 1 | | | TABLE OF CONTENTS:
.. 67      0130 1 | | | -
.. 68      0131 1 | | |
.. 69      0132 1 | | FORWARD ROUTINE
.. 70      0133 1 | | | RPG$IOEXCEPTION : NOVALUE ;
.. 71      0134 1 | | |
.. 72      0135 1 | | | +
.. 73      0136 1 | | | INCLUDE FILES
.. 74      0137 1 | | | NONE
.. 75      0138 1 | | | -
.. 76      0139 1 | | |
.. 77      0140 1 | | | +
.. 78      0141 1 | | | MACROS
.. 79      0142 1 | | | -
.. 80      0143 1 | | |
.. 81      0144 1 | | MACRO
.. 82      0145 1 | | | RETRIEVE = 0,0,2,0%;
.. 83      0146 1 | | |
.. 84      0147 1 | | | +
.. 85      0148 1 | | | EXTERNAL REFERENCES
.. 86      0149 1 | | | -
.. 87      0150 1 | | |
.. 88      0151 1 | | EXTERNAL ROUTINE
.. 89      0152 1 | | | LIB$STOP:NOVALUE;
.. 90      0153 1 | | |
.. 91      0154 1 | | EXTERNAL LITERAL
.. 92      0155 1 | | | RPG$_ENDOF_FIL,
.. 93      0156 1 | | | RPG$_EXTINDOFF,
.. 94      0157 1 | | | RPG$_FATINTERR,
.. 95      0158 1 | | | RPG$_FORFILE,
.. 96      0159 1 | | | RPG$_KEYNOTFND,
.. 97      0160 1 | | | RPG$_KEYNOTMAT,
.. 98      0161 1 | | | RPG$_NONASCKEY,
.. 99      0162 1 | | | RPG$_UPDNO_CUR;
.. 100     0163 1 | | |
```

! Signal fatal error

! End of file error
! File not open error
! Fatal internal error
! for file 'filename' error
! Key not found error
! Key doesn't match error
! Out of sequence error
! No current record error


```

: 102      0164 1 %SBTTL 'RPG$IOEXCEPTION - Signal I/O errors'
: 103      0165 1 GLOBAL ROUTINE RPG$IOEXCEPTION (
: 104      0166 1
: 105      0167 1          FLAGS:          BLOCK[,BYTE],      ! Error value location indicator
: 106      0168 1          RAB:          REF BLOCK[,BYTE]      ! Address of RAB
: 107      0169 1          ): NOVALUE=
: 108      0170 1
: 109      0171 1      ++
: 110      0172 1      FUNCTIONAL DESCRIPTION:
: 111      0173 1
: 112      0174 1          This procedure is called to process a wide variety of I/O
: 113      0175 1      exceptions. It retrieves the file name string and signals the errors.
: 114      0176 1
: 115      0177 1      CALLING SEQUENCE:
: 116      0178 1
: 117      0179 1          CALL RPG$IOEXCEPTION (flags.rl.v, rab.rr.r)
: 118      0180 1
: 119      0181 1      FORMAL PARAMETERS:
: 120      0182 1
: 121      0183 1          flags          - Indication of where to retrieve the condition value
: 122      0184 1
: 123      0185 1          rab            - Address of RAB
: 124      0186 1
: 125      0187 1
: 126      0188 1      IMPLICIT INPUTS:
: 127      0189 1
: 128      0190 1          NONE
: 129      0191 1
: 130      0192 1      IMPLICIT OUTPUTS:
: 131      0193 1
: 132      0194 1          NONE
: 133      0195 1
: 134      0196 1      ROUTINE VALUE:
: 135      0197 1
: 136      0198 1          NONE
: 137      0199 1
: 138      0200 1      SIDE EFFECTS:
: 139      0201 1
: 140      0202 1          NONE
: 141      0203 1
: 142      0204 1      --

```

```

144 0205 2 BEGIN
145 0206 2
146 0207 2 LOCAL
147 0208 2 CONDITION,
148 0209 2 FAB: REF BLOCK[,BYTE],
149 0210 2 NAM: REF BLOCK[,BYTE],
150 0211 2
151 0212 2
152 0213 2 RMS_STS,
153 0214 2 RMS_STV,
154 0215 2 RSADESC: VECTOR[2];
155 0216 2
156 0217 2
157 0218 2 !+ Ensure that the RMS structures are valid.
158 0219 2
159 0220 2 IF .RAB EQL 0
160 0221 2 THEN
161 0222 2 LIB$STOP (RPG$_FATINTERR);
162 0223 2
163 0224 2 FAB = .RAB[RAB$_FAB];
164 0225 2 IF .FAB EQL 0
165 0226 2 THEN
166 0227 2 LIB$STOP (RPG$_FATINTERR);
167 0228 2
168 0229 2 NAM = .FAB[FAB$_NAM];
169 0230 2 IF .NAM EQL 0
170 0231 2 THEN
171 0232 2 LIB$STOP (RPG$_FATINTERR);
172 0233 2
173 0234 2 !+
174 0235 2 ! Fetch the address and length of the RSA as defined in the NAM block.
175 0236 2 ! If the length field is zero, then fetch the extended string length field.
176 0237 2 ! If that too has a zero length, then just use the FAB file name string
177 0238 2 ! name and string length.
178 0239 2
179 0240 2
180 0241 2 RSADESC[0] = .NAM[NAM$_RSL];
181 0242 2 IF .RSADESC[0] EQL 0
182 0243 2 THEN
183 0244 2 BEGIN
184 0245 2
185 0246 2 RSADESC[0] = .NAM[NAM$_ESL];
186 0247 2 IF .RSADESC[0] EQL 0
187 0248 2 THEN
188 0249 2 BEGIN
189 0250 2
190 0251 2 RSADESC[0] = .FAB[FAB$_DNS];
191 0252 2 RSADESC[1] = .FAB[FAB$_DNA];
192 0253 2
193 0254 2 END
194 0255 2 ELSE
195 0256 2 RSADESC[1] = .NAM[NAM$_ESA];
196 0257 2
197 0258 2 END
198 0259 2 ELSE
199 0260 2 RSADESC[1] = .NAM[NAM$_RSA];
200 0261 2

```

! Longword RPG\$ error number
! Address of FAB
! Address of NAM block-used to
! communicate optional filename
! related information.
! RMS STS value
! RMS STV value
! Descriptor for RSA (resultant string area)

! Error if RAB addr = 0

! Fetch FAB address

! Error if FAB addr = 0

! Fetch addr of NAM block

! Error if addr of NAM block = 0

! Resultant string length

! Expanded string length

! Default file spec string length
! Default file spec string addr

! Expanded string addr

! Resultant string addr


```
201 0262 2
202 0263 2
203 0264 2
204 0265 2
205 0266 2
206 0267 2
207 0268 2
208 0269 2
209 0270 2
210 0271 2
211 0272 2
212 0273 2
213 0274 3
214 0275 3
215 0276 3
216 0277 3
217 0278 3
218 0279 3
219 0280 3
220 0281 3
221 0282 4
222 0283 4
223 0284 4
224 0285 4
225 0286 4
226 0287 4
227 0288 4
228 0289 4
229 0290 4
230 0291 4
231 0292 4
232 0293 4
233 0294 4
234 0295 3
235 0296 4
236 0297 4
237 0298 4
238 0299 4
239 0300 4
240 0301 4
241 0302 4
242 0303 4
243 0304 4
244 0305 4
245 0306 4
246 0307 4
247 0308 3
248 0309 3
249 0310 3
250 0311 3
251 0312 3
252 0313 3
253 0314 4
254 0315 4
255 0316 4
256 0317 4
257 0318 4

!+
!- Case on retrieve indicator.
CASE .FLAGS[RETRIEVE] FROM RPG$K_ERR_RAB TO RPG$K_ERR_FILE_CNTXT OF
  SET
    [OUTRANGE]: ! Invalid value
      LIB$STOP (RPG$FATINTERR);
    [RPG$K_ERR_RAB, RPG$K_ERR_FAB]: ! RMS RAB/FAB error
      BEGIN
        !+
        !- RMS error. Pick up the appropriate STS and STV.
        IF .FLAGS[RETRIEVE] EQL RPG$K_ERR_RAB ! Check specifically for RAB error
          THEN
            BEGIN
              !+
              !- Here we know that we have a RAB error.
              IF .RAB[RAB$L_STS] EQL 0 !\If status of RAB = 0
                THEN ! then signal as a
                  LIB$STOP (RPG$FATINTERR); ! /fatal internal error.
                  RMS_STS = .RAB[RAB$L_STS]; ! Load status code from RAB
                  RMS_STV = .RAB[RAB$L_STV]; ! Load status value from RAB
            END
          ELSE
            BEGIN
              !+
              !- Here we know that we have a FAB error.
              IF .FAB[FAB$L_STS] EQL 0 !\If status of FAB = 0,
                THEN ! then signal a
                  LIB$STOP (RPG$FATINTERR); ! /fatal internal error.
                  RMS_STS = .FAB[FAB$L_STS]; ! Load status code from FAB
                  RMS_STV = .FAB[FAB$L_STV]; ! Load status value from FAB
            END;
          !+
          !- Map the STS value into the appropriate RPG file status.
          CONDITION = (SELECTONE .RMS_STS OF
            SET
              [RMS$_EOF]: RPG$_ENDOF_FIL;
```


RPG\$IOEXCEPTION
1-003

Signal I/O errors
RPG\$IOEXCEPTION - Signal I/O errors

C 15
16-Sep-1984 02:16:13
14-Sep-1984 13:04:19

VAX-11 Bliss-32 V4.0-742
[RPGRTL.SRC:RPGIOEXCE.B32;1

Page 6
(4)

```
: 258      0319  4      [RMS$_ISI]:      RPG$_EXTINDOFF;
: 259      0320  4
: 260      0321  4      [RMS$_REF]:      RPG$_KEYNOTMAT;
: 261      0322  4
: 262      0323  4      [RMS$_RNF]:      RPG$_KEYNOTFND;
: 263      0324  4
: 264      0325  4      [RMS$_SEQ]:      RPG$_NONASCKEY;
: 265      0326  4
: 266      0327  4      [RMS$_CUR]:      RPG$_UPDNO_CUR;
: 267      0328  4
: 268      0329  4      [OTHERWISE]:      0;
: 269      0330  4
: 270      0331  3      TES);
: 271      0332  3
: 272      0333  3      !+
: 273      0334  3      !- Signal the error.
: 274      0335  3
: 275      0336  3
: 276      0337  3      IF .CONDITION NEQ 0
: 277      0338  3      THEN
: 278      0339  3          LIB$STOP (.CONDITION, 0, RPG$_FORFILE, 1, RSADESC)
: 279      0340  3      ELSE
: 280      0341  3          LIB$STOP (.RMS_STS, .RMS_STV, RPG$_FORFILE, 1, RSADESC);
: 281      0342  3
: 282      0343  2      END;
: 283      0344  2
: 284      0345  2      [RPG$_ERR_FILE_CNTXT]:
: 285      0346  2
: 286      0347  2      !+
: 287      0348  2      !- Fetch condition value from the RPG$_CTX_ERROR field
: 288      0349  2      and signal the error.
: 289      0350  2
: 290      0351  2
: 291      0352  2      LIB$STOP (.RAB[RPG$_CTX_ERROR], 0, RPG$_FORFILE, 1, RSADESC);
: 292      0353  2
: 293      0354  2      TES;
: 294      0355  2
: 295      0356  1      END;
```

.TITLE RPG\$IOEXCEPTION Signal I/O errors
.IDENT \1-003\

.EXTRN LIB\$STOP, RPG\$_ENDOF_FIL
.EXTRN RPG\$_EXTINDOFF, RPG\$_FATINTERR
.EXTRN RPG\$_FORFILE, RPG\$_KEYNOTFND
.EXTRN RPG\$_KEYNOTMAT, RPG\$_NONASCKEY
.EXTRN RPG\$_UPDNO_CUR

.PSECT _RPG\$CODE, NOWRT, SHR, PIC, 2

.ENTRY RPG\$IOEXCEPTION, Save R2,R3,R4,R5,R6,R7,R8 ; 0165
MOVL #RPG\$_FORFILE, R8 ;
MOVL #RPG\$_FATINTERR, R7 ;
MOVAB LIB\$STOP, R6 ;
SUBL2 #8, SP ;
MOVL RAB, R4 ; 0220

01FC 00000
58 0000C000G 8F D0 00002
57 00000000G 8F D0 00009
56 00000000G 00 9E 00010
5E 08 C2 00017
54 08 AC D0 0001A

			05	12	0001E	BNEQ	1\$			
			57	DD	00020	PUSHL	R7		0222	
	66		01	FB	00022	CALLS	#1, LIB\$STOP			
	53	3C	A4	D0	00025	1\$:	MOVL	60(R4), FAB	0224	
			05	12	00029	BNEQ	2\$		0225	
			57	DD	0002B	PUSHL	R7		0227	
	66		01	FB	0002D	CALLS	#1, LIB\$STOP			
	52	28	A3	D0	00030	2\$:	MOVL	40(FAB), NAM	0229	
			05	12	00034	BNEQ	3\$		0230	
			57	DD	00036	PUSHL	R7		0232	
	66		01	FB	00038	CALLS	#1, LIB\$STOP			
	6E	03	A2	9A	0003B	3\$:	MOVZBL	3(NAM), RSADESC	0241	
			18	12	0003F	BNEQ	5\$		0242	
	6E	0B	A2	9A	00041	MOVZBL	11(NAM), RSADESC		0246	
			0B	12	00045	BNEQ	4\$		0247	
	6E	35	A3	9A	00047	MOVZBL	53(FAB), RSADESC		0251	
	04	AE	30	A3	D0	0004B	MOVL	48(FAB), RSADESC+4	0252	
			0C	11	00050	BRB	6\$		0247	
	04	AE	0C	A2	D0	00052	4\$:	MOVL	12(NAM), RSADESC+4	0256
			05	11	00057	BRB	6\$		0242	
	04	AE	04	A2	D0	00059	5\$:	MOVL	4(NAM), RSADESC+4	0260
55			00	EF	0005E	6\$:	EXTZV	#0, #2, FLAGS, R5	0267	
	04	AC	01	55	CF	00064	CASEL	R5, #1, #2		
	00B8		000C		00068	7\$:	.WORD	8\$-7\$, -		
								8\$-7\$, -		
								21\$-7\$		
			57	DD	0006E	PUSHL	R7		0271	
	66		01	FB	00070	CALLS	#1, LIB\$STOP			
				04	00073	RET				
01			00	ED	00074	8\$:	CMPZV	#0, #2, FLAGS, #1	0280	
	02		10	12	0007A	BNEQ	10\$			
		08	A4	D5	0007C	TSTL	8(R4)		0288	
			05	12	0007F	BNEQ	9\$			
			57	DD	00081	PUSHL	R7		0290	
	66		01	FB	00083	CALLS	#1, LIB\$STOP			
	51	08	A4	7D	00086	9\$:	MOVQ	8(R4), RMS_STS	0291	
			0E	11	0008A	BRB	12\$		0280	
		08	A3	D5	0008C	10\$:	TSTL	8(FAB)	0302	
			05	12	0008F	BNEQ	11\$			
			57	DD	00091	PUSHL	R7		0304	
	66		01	FB	00093	CALLS	#1, LIB\$STOP			
	51	08	A3	7D	00096	11\$:	MOVQ	8(FAB), RMS_STS	0305	
	8F		51	D1	0009A	12\$:	CMPL	RMS_STS, #98938	0317	
0001827A			09	12	000A1	BNEQ	13\$			
	50	00000000G	8F	D0	000A3	MOVL	#RPG\$_ENDOF_FIL, CONDITION			
			5C	11	000AA	BRB	19\$			
00018584	8F		51	D1	000AC	13\$:	CMPL	RMS_STS, #99716	0319	
			09	12	000B3	BNEQ	14\$			
	50	00000000G	8F	D0	000B5	MOVL	#RPG\$_EXTINDOFF, CONDITION			
			4A	11	000BC	BRB	19\$			
0001875C	8F		51	D1	000BE	14\$:	CMPL	RMS_STS, #100188	0321	
			09	12	000C5	BNEQ	15\$			
	50	00000000G	8F	D0	000C7	MOVL	#RPG\$_KEYNOTMAT, CONDITION			
			38	11	000CE	BRB	19\$			
000182B2	8F		51	D1	000D0	15\$:	CMPL	RMS_STS, #98994	0323	
			09	12	000D7	BNEQ	16\$			
	50	00000000G	8F	D0	000D9	MOVL	#RPG\$_KEYNOTFND, CONDITION			

000186AC	8F	26	11	000E0	BRB	19\$		
		51	D1	000E2	16\$:	CMPL	RMS_STS, #100012	0325
	50 00000000G	09	12	000E9	BNEQ	17\$		
		8F	D0	000EB	MOVL	#RPG\$_NONASCKEY, CONDITION		
000184B4	8F	14	11	000F2	BRB	19\$		
		51	D1	000F4	17\$:	CMPL	RMS_STS, #99508	0327
	50 00000000G	09	12	000FB	BNEQ	18\$		
		8F	D0	000FD	MOVL	#RPG\$_UPDNO_CUR, CONDITION		
		02	11	00104	BRB	19\$		
		50	D4	00106	18\$:	CLRL	CONDITION	0329
		0C	13	00108	19\$:	BEQL	20\$	0337
		5E	DD	0010A	PUSHL	SP		0339
		01	DD	0010C	PUSHL	#1		
		58	DD	0010E	PUSHL	R8		
		7E	D4	00110	CLRL	-(SP)		
		50	DD	00112	PUSHL	CONDITION		
		15	11	00114	BRB	22\$		
		5E	DD	00116	20\$:	PUSHL	SP	0341
		01	DD	00118	PUSHL	#1		
	0106	8F	BB	0011A	PUSHR	#^M<R1,R2,R8>		
		0B	11	0011E	BRB	22\$		
		5E	DD	00120	21\$:	PUSHL	SP	0352
		01	DD	00122	PUSHL	#1		
		58	DD	00124	PUSHL	R8		
		7E	D4	00126	CLRL	-(SP)		
	F8	A4	DD	00128	PUSHL	-8(R4)		
	66	05	FB	0012B	22\$:	CALLS	#5, LIB\$STOP	
		04	0012E	RET				0356

; Routine Size: 303 bytes,

Routine Base: _RPG\$CODE + 0000

; 296

; 297

0357 1

0358 0 END ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
_RPG\$CODE	303	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	18	0	581	00:01.0
-\$255\$DUA28:[RPGR TL.OBJ]RPGLIB.L32;1	54	4	7	9	00:00.1

RPG\$IOEXCEPTION Signal I/O errors
1-003 RPG\$IOEXCEPTION - Signal I/O errors

F 15
16-Sep-1984 02:16:13
14-Sep-1984 13:04:19

VAX-11 Bliss-32 V4.0-742
[RPGRTL.SRC]RPGIOEXCE.B32;1

Page 9
(4)

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:RPGIOEXCE/OBJ=OBJ\$:RPGIOEXCE MSRC\$:RPGIOEXCE/UPDATE=(ENH\$:RPGIOEXCE)

: Size: 303 code + 0 data bytes
: Run Time: 00:07.1
: Elapsed Time: 00:24.9
: Lines/CPU Min: 3038
: Lexemes/CPU-Min: 12857
: Memory Used: 92 pages
: Compilation Complete

0331

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY